UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

LAGRANGIAN RELAXATION FOR ROUTING WITH TIME WINDOWS

by

Jacques Desrosiers, François Soumis
and Michel Sauvé

Présenté à la conférence "IFORS 84", Washington, D.C., U.S.A.,
6-10 août 1984

Ce rapport a été publié grâce à une subvention du fonds F.C.A.C.
pour l'aide et le soutien à la recherche.

Université de Montréal
Centre de recherche sur les transports - Publication #337
École des Hautes Études Commerciales - Cahier G-84-03
Février 1984

Dépôt légal - 2e trimestre 1984

LAGRANGIAN RELAXATION FOR ROUTING WITH TIME WINDOWS

La relaxation lagrangienne pour le routage sous contraintes d'horaires

JACQUES DESROSIERS
FRANCOIS SOUMIS
MICHEL SAUVE

GERAD, Ecole des Hautes Etudes Commerciales de Montréal, Canada

Abstract. In this article, we examine the use of Lagrangian relaxation to obtain a lower bound on the minimum fleet size for the m-TSP with time window constraints on the nodes. The constraints on visiting each node are relaxed. The subgradient method was found to be inefficient while an augmented Lagrangian approach gave satisfactory numerical results when compared with a primal column generation algorithm.

Résumé. Dans cet article, nous étudions l'utilisation de la relaxation lagrangienne pour l'obtention d'une borne inférieure sur la taille minimale de la flotte pour le problème du m-voyageur de commerce avec contraintes d'intervalles de temps sur le noeuds. La relaxation est faite pour les contraintes de visites aux noeuds. La méthode des sous-gradients s'est avérée inefficace alors que la méthode du lagrangien augmenté fournit des résultats numériques satisfaisants. La comparaison est faite avec un algorithme primal de génération de colonnes.

LAGRANGIAN RELAXATION FOR ROUTING WITH TIME WINDOWS

1. INTRODUCTION

In this paper, we examine the use of Lagrangian relaxation to minimize the fleet size for the m-TSP with time window constraints on the nodes. The problem formulation involves flow variables x_{ij} and n continuous time variables $t_i \in [a_i, b_i]$ associated with each node. Let index i = 0 (origin) or j = 0 (destination) denote the single depot, while t_{ij}, i,j = 0,1,...,n represent travel times between the nodes of the network. An arc (i,j), i,j = 1,...,n, exists if $a_i + t_{ij} \leq b_j$. We assume that $t_{ii} = \infty$, i = 1,...,n and $t_{00} = 0$. The optimal fleet size respecting the scheduling constraints is given by the solution of the integer linear programming problem IP:

$$Z_{IP} = \min \sum_{j=1}^{n} x_{0j} \tag{1}$$

subject to:

$$\sum_{j=0}^{n} x_{ij} = 1 \qquad i = 1,\ldots,n \tag{2}$$

$$\sum_{i=0}^{n} x_{ik} = \sum_{j=0}^{n} x_{kj} \qquad k = 1,\ldots,n \tag{3}$$

$$\sum_{j=0}^{n} x_{0j} = n \tag{4}$$

$$\sum_{i=0}^{n} x_{i0} = n \tag{5}$$

$$t_i + t_{ij} - t_j \leq M_{ij}(1-x_{ij}) \qquad i,j = 1,\ldots,n \tag{6}$$

$$a_i \leq t_i \leq b_i \qquad i = 1,\ldots,n \tag{7}$$

$$x_{ij} = 0, \text{ integer} \qquad i,j = 0,1,\ldots,n \tag{8}$$

where $M_{ij} = b_i + t_{ij} - a_j \qquad i,j = 1,\ldots,n$

Relations (1-5,8) form a simple network flow problem N without scheduling constraints. Relations (6) describe the compatibility requirements between routes and schedules and also ensure subtour elimination, while relations (7) establish the time windows within which nodes must be visited. Constraints (4) and (5) are obtained from relations (2) and (3) by defining $x_{00} = \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij}$; they are thus redundant in this formulation but become operative if constraints (2) are relaxed.

Two primal algorithms have been developed to solve this problem. The first uses column generation [3] on a set partitioning problem made up of feasible routes. The linear version of the problem is solved by the simplex method; the feasible routes are solutions of a shortest path problem with scheduling constraints (2) and an optimal integer solution is obtained using a branch-and-bound

tree. The second algorithm solves a network relaxation problem (4) at each node of a branch-and-bound tree. In this problem, we present a dual method wich gives a lower bound as good as that provided by the column generation method, but is more economical in terms of time and memory. This bound can be intenrated into the second algorithm.

2. BASIC CONSTRUCTIONS OF LAGRANGIAN RELAXATION

Let IP denote an integer linear programming problem: Z_{IP}= min c'x subject to: $Ax = b$ (or $\geq b$), $x \in \underline{X}$, x integer where $\underline{X}$ is a set of linear constraints. LP is the linear version of IP, and Z_{LP} is its optimal value. The lagrangian relaxation of IP relative to $Ax = b$ (or $\geq b$) with a vector $\lambda \in R$ (or $\lambda \geq 0$) is:

$$Z_{L(\lambda)} = \min c'x + \lambda'(b - Ax) \quad \text{subject to:} \quad x \in \underline{X}, \ x \text{ integer.}$$

It is well known (6,7) that $Z_{L(\lambda)} \leq Z_{IP}$ and that the best choice for λ would be an optimal solution to $Z_L = \max_{\lambda} Z_{L(\lambda)}$. Problem L is called the Lagrangian dual and its optimal value lies between Z_{LP} and Z_{IP}: $Z_{LP} \leq Z_L \leq Z_{IP}$. Finally, if IP is feasible and the optimal value of L λ) is not altered by dropping the integrality constraints on x (Integrality Property), then $Z_{LP} = Z_L$.

3. LAGRANGIAN RELAXATIONS OF THE m-TSP WITH TIME WINDOWS

Two Lagrangian relaxations have been studied for this problem (5). The first one is obtained by the relaxation of the compatibility constraints (6) between the flow variables and the time variables. The problem $L(\lambda)$ with $\lambda = (\lambda_{ij} \geq 0, \ i,j = 1,\ldots,n)$ can be decomposed into the network problem N on the flow variables and a trivial problem on the time variables. Problem $L(\lambda)$ therefore has the integrality property and $Z_{LP} = Z_L$. Furthermore, the problem LP is the network problem N with additional constraints (6-7) and without integrality requirements, these constraints are generally inactive. In most cases, $Z_N = Z_{LP}$. There is thus nothing to be gained by using this Lagrangian relaxation. Numerical results obtained when optimizing school transportation fleet sizes using the network relaxation indicates that the bound Z_N is of poor quality: ave $Z_N = 0.928 Z_{IP}$ which often represents a significant gap of 5 to 10 vehicles.

The Lagrangian relaxation of constraints (2) on visiting each node generates problems $L(\lambda)$ with $\lambda = (\lambda_i \in R, \ i = 1,\ldots,n)$:

LAGRANGIAN RELAXATION FOR ROUTING WITH TIME WINDOWS

$$Z_{L(\lambda)} = \min \sum_{j=1}^{n} x_{0j} + \sum_{i=1}^{n} \lambda_i \left(1 - \sum_{j=0}^{n} x_{ij}\right) \text{ subject to: (3-8)}$$

For a given λ, $L(\lambda)$ is a shortest path problem with time windows on the nodes (SPTW) with the special feature that n units of flow must follow the shortest path from the depot origin to the depot destination. The SPTW problem does not possess the integrality property but an optimal integer solution is easily obtained using dynamic programming (2).

The generalized linear programming algorithm (9) is a primal method for solving $Z_L = \max_{\lambda} Z_{L(\lambda)}$. We showed in (5) that the problem L is equivalent to the Dantzig-Wolfe decomposition of the original problem obtained by placing (2) in the master problem and solving the subproblem (3-8) in integer form. The master problem is then a set partitioning problem without integrality constraints (linear set partitioning LSP) while the subproblem is a SPTW problem with integrality requirements. As $Z_L = Z_{LSP}$, the evaluation of this Dantzig-Wolfe decomposition which was carried out in (3) provides information about Z_L. The quality of this bound is excellent: ave $Z_{LSP} = 0.992 Z_{IP}$. The bound is further improved when it is rounded up to the next integer. This procedure gives the optimal fleet size for all problems examined. The excellent quality of this bound is essentially due to the integer solution of the SPTW problem. The results of Fisher (6) who solve knapsack subproblems to handle the generalized assignment problem confirm the value of this approach.

This Lagrangian relaxation can be further improved by retaining the constraints which do not destroy the shortest path structure. Given LB, a lower bound on the fleet size (obtained for example using the network relaxation N) and UB, an upper bound (a feasible solution), we introduce constraints

$$LB \leq \sum_{j=1}^{n} x_{0j} \leq UB. \qquad (9\text{-}10)$$

This relaxation is also improved by not relaxing the constraints (2) on visiting each node for the subset S of nodes which have only the destination depot node as successor: $S = \{i \mid a_i + t_{ij} > b_j, j = 1,\ldots,n\}$. Each node $i \in S$ is considered to be a pure destination for the SPTW problem. It is sufficient to remove arcs $(i,0)$, $\forall i \in S$ from the network. One unit of flow is then applied to each path leading from the origin depot to $i \in S$, in the shortest path tree. On the path from the origin depot to the destination depot, the flow is set at the maximum value (UB - |S| units) if the length is negative, and at the minimum value (LB - |S| units) if the length is positive. In the latter case, the remaining flow is assigned to the arc (0,0) at zero cost.

LAGRANGIAN RELAXATION FOR ROUTING WITH TIME WINDOWS

We first discuss the results obtained using subgradient methods, and then those achieved with an augmented Lagrangian method. Tests were carried out on the following Lagrangian relaxation:

$$Z_{L(\lambda)} = \min \sum_{j=1}^{n} x_{0j} + \sum_{i \in \bar{S}} \lambda_i \left(1 - \sum_{j=0}^{n} x_{ij}\right)$$

subject to: (3) - (8)

$$LB \leq \sum_{j=1}^{n} x_{0j} \leq UB \tag{9-10}$$

$$\sum_{j=0}^{n} x_{ij} = 1 \qquad i \in S \tag{11}$$

To simplify the notation, we define

$$x_i = \sum_{j=0}^{n} x_{ij};$$

x_i represents the total flow passing through node i. The objective function then becomes:

$$\min \sum_{j=1}^{n} x_{0j} + \sum_{i \in \bar{S}} \lambda_i (1 - x_i)$$

$$= \min \sum_{j=1}^{n} x_{0j} - \sum_{i \in \bar{S}} \lambda_i x_i + \sum_{i \in \bar{S}} \lambda_i \tag{12}$$

4. THE SUBGRADIENT METHOD

The main advantage of the subgradient method [8] is its simplicity. At iteration k, let $\lambda^k = (\lambda_i^k \in R,\ i \in \bar{S})$ and let $x^k = (\sum_{j=1}^{n} x_{0j}^k;\ x_i^k,\ i \in \bar{S})$ be an optimal solution of problem $L(\lambda^k)$. $Z_{L(\lambda^k)}$ is a lower bound on Z_{IP} and the vector λ^{k+1} is calculated as follows:

$$\lambda_i^{k+1} = \lambda_i^k + r^k (1 - x_i^k) \qquad i \in \bar{S} \tag{13}$$

where $r^k = \dfrac{r^k (UB - L(\lambda^k))}{\sum_{i \in \bar{S}} (1 - x_i^k)^2}$ (Polyak's step) (14)

with $0 \leq r^k \leq 2$, $\sum_k r^k \to \infty$, $r^k \to 0$.

If the lower bound LB is too low, this is quickly determined. In fact, if $Z_{L(\lambda)} > LB$ it is impossible to construct an integer solution with LB vehicles so the lower bound must be increased by $\lceil Z_{L(\lambda)} - LB \rceil$ units.

This method was thoroughly tested with and without the supplementary constraints (9-11). These constraints produced less extreme directions and increased the value of the objective function. They reduced computation time by half when the upper bound UB was set at the optimal value (known). However, if the upper bound is too high, the choice of step becomes very difficult: one vehicle too many in (14) changes the multipliers too much and requires a rapidly diminishing step. The cost structure produced a very ill-conditioned problem (leaps in the objective function) and it is very difficult to establish a suitable stopping rule for the algorithm.

5. THE AUGMENTED LAGRANGIAN METHOD

The augmented Lagrangian method [1] is more difficult to use than the subgradient approach but it provides more information. For a penalty factor $c > 0$ and multipliers $\lambda = (\lambda_i \in R,\ i \in \bar{S})$, the following quadratic function is to be minimized:

$$L_c(X,\lambda) = \sum_{j=1}^{n} x_{0j} + \sum_{i\in\bar{S}} \lambda_i\Big(1 - \sum_{j=0}^{n} x_{ij}\Big) + \frac{c}{2} \sum_{i\in\bar{S}} \Big(1 - \sum_{j=0}^{n} x_{ij}\Big)^2$$

$$= \sum_{j=1}^{n} x_{0j} + \sum_{i\in\bar{S}} \lambda_i(1 - x_i) + \frac{c}{2} \sum_{i\in\bar{S}} (1 - x_i)^2 \qquad (15)$$

Using the notation

$Z_{L_c}(\lambda) = \min L_c(X,\lambda)$ subject to (3-11) and $Z_{L_c} = \max_{\lambda} Z_{L_c}(\lambda)$, we have that $Z_{L_c} \le Z_{IP}$.

5.1 ADJUSTMENT OF THE MULTIPLIERS

Let λ^k and c be fixed and let $X^k = \Big(\sum_{j=1}^{n} x_{0j}^k ;\ x_i^k,\ i \in \bar{S}\Big)$ be an optimal solution of the quadratic problem $L_c(\lambda^k)$; then the new multipliers are calculated as follows:

$$\lambda_i^{k+1} = \lambda_i^k + c(1 - x_i^k), \qquad i \in \bar{S} \qquad (16)$$

Note that the penalty factor c may vary freely from one iteration to the next, but when c is too small, the effect of the penalty term is reduced, and when c is too large, the quadratic problem $L_c(\lambda)$ is difficult to solve. In our application, the quadratic problem is not solved optimally but is approximated rather roughly: the vector X has no interest other than to provide a direction indicator

for the adjustment of the multipliers.

5.2 SOLUTION OF THE QUADRATIC PROBLEM

To take advantage of the special structure of constraints (3-11), the problem $L_c(\lambda^k)$ is solved using the Frank-Wolfe algorithm.

I) Let X^ℓ be a feasible flow vector with respect to constraints (3-11) at iteration ℓ of the Frank-Wolfe algorithm. An extreme point is calculated by taking the linear approximation in X^ℓ of the quadratic objective function i.e. $\nabla_x L_c(X^\ell, \lambda^k) \cdot Y$. The approximation subproblem is the following SPTW problem:

$$\min \sum_{j=1}^{n} y_{0j} - \sum_{i \in \bar{S}} \left[\lambda_i^k + c(1 - \sum_{j=0}^{n} x_{ij})\right] \left(\sum_{j=0}^{n} y_{ij}\right) \quad \text{subject to: (3-11)}$$

$$= \min \sum_{j=1}^{n} y_{0j} - \sum_{i \in \bar{S}} \left[\lambda_i^k + c(1 - x_i^\ell)\right] y_i \qquad \text{subject to : (3-11)}$$

Let Y^ℓ be an optimal solution to this problem; note that this solution is not changed if we add the constant $\sum_{i \in \bar{S}} \left[\lambda_i^k + c(1 - x_i^\ell)\right]$ to the objective function. With this additional constant, the subproblem to be solved is the Lagrangian problem $L(\lambda^k + c(1 - x^\ell))$ where $x^\ell = (x_i^\ell, i \in \bar{S})$. This gives a lower bound on Z_{IP} at each solution of the sub-problem.

II) The Frank-Wolfe algorithm stops when $\nabla_x L_c(X^\ell, \lambda^k) \cdot (Y^\ell - X^k) < \varepsilon_{FW}$ i.e. when $\delta_{FW} = \sum_{j=1}^{n} (y_{0j}^\ell - x_{0j}^\ell) - \sum_{i \in \bar{S}} \left[\lambda_i^k + c(1 - x_i^\ell)\right] (y_i^\ell - x_i^k) < \varepsilon_{FW}$. (17)

We then set $X^k := X^\ell$ to adjust the multipliers λ. As the function $L_c(X,\lambda)$ is convex in X, it can be shown that

$$L_c(X^\ell, \lambda^k) + \delta^\ell \leq Z_{L_c}(\lambda^k) \leq Z_{IP}$$

which gives a second lower bound on Z_{IP} at each solution of the subproblem.

III) When the stopping rule is not satisfied, the optimal step in the descent direction $Y^\ell - X^\ell$ must be found as follows:

$$\min_{0 \leq \theta \leq 1} L_c((1-\theta)X^\ell + \theta Y^\ell, \lambda^k)$$

Analytically, we obtain by differentiating with respect to θ:

$$\theta^{\ell} = \min \left\{ 1, \frac{-\delta^{\ell}}{c \sum_{i \in \bar{S}} (y_i^{\ell} - x_i^{\ell})^2} \right\} \tag{18}$$

The new flow vector becomes:

$X^{\ell+1} = (1-\theta^{\ell})\, X^{\ell} + \theta^{\ell} Y^{\ell}$ and we return to 1) to continue optimization of the quadratic problem $L_c(\lambda^k)$.

5.3 TIGHTENING OF BOUNDS LB AND UB

At each solution of the SPTW problem, two lower bounds denoted by LB_1 and LB_2 are produced. If (LB_1 or LB_2) > LB, then the lower bound is increased to $LB' = \lceil \max(LB_1, LB_2) \rceil$. The current flow vector X^{ℓ} with cost $\sum_{j=1}^{n} x_{0j}^{\ell} \notin [LB, UB]$ usually falls outside the new polyhedron. In this case, we project this vector into the new polyhedron by increasing its cost by (LB'-LB) units and by proportionally distributing the additional units of flow over all nodes $i \in \bar{S}$ for which $x_i < 1$, i.e.

$$x_i^{\ell} := x_i^{\ell} + (LB'-LB) \frac{\max\,(0,\, 1-x_i^{\ell})}{\sum_{i \in \bar{S}} \max(0,\, 1-x_i^{\ell})}, \quad i \in \bar{S} \tag{19}$$

The upper bound is also improved when

$$\left\lceil \sum_{j=1}^{n} x_{0j}^{\ell} + \sum_{i \in \bar{S}} \max\,(0,\, 1-x_i^{\ell}) \right\rceil < UB \tag{20}$$

i.e. by calculating the maximum flow required to cover all the nodes directly from the origin depot.

Figure 1 describes the main points of the algorithm with the following stopping rule operating on the optimization of the augmented Lagrangian.

$$\delta = \sum_{i \in \bar{S}} \max(0,\, 1-x_i^{k}) < \varepsilon. \tag{21}$$

LAGRANGIAN RELAXATION FOR ROUTING WITH TIME WINDOWS

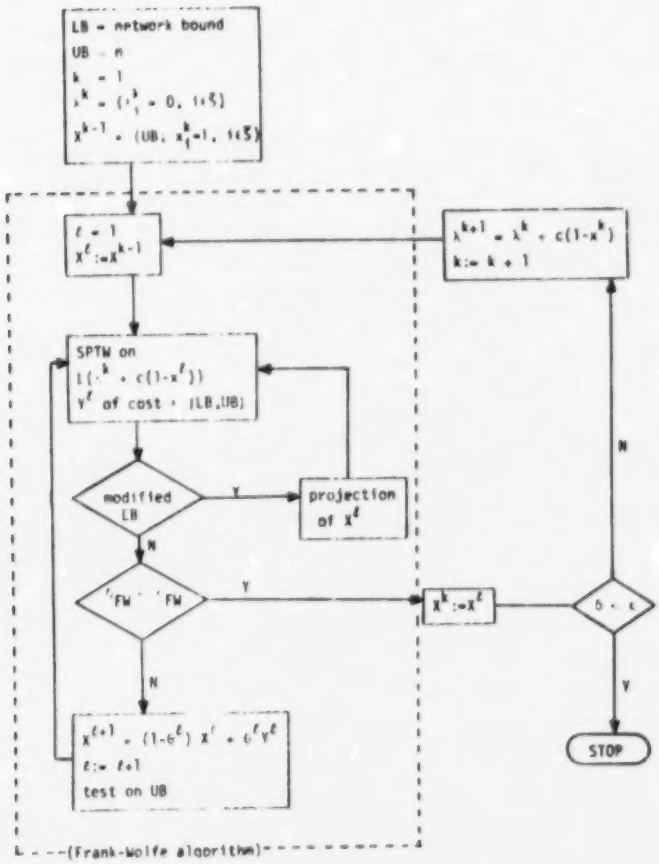

Figure 1: Augmented Lagrangian procedure

6. NUMERICAL RESULTS AND CONCLUSIONS

The algorithm was evaluated on 28 school bus transportation problems (3). Each problem is identified by a trip number n (48, 100, 151) followed by AM or PM to indicate whether morning, or afternoon transportation is being considered, plus a constant Δ replacing the original intervals (a_i, b_i) by $(a_i, b_i+\Delta)$. The following parameters were used in the solution: LB = network relaxation bound, UB = n, $\epsilon_{FW} = .1$, $c = .1$ and $\epsilon = 1$. The vector λ is initially zero, but at the end of the first iteration of the Frank-Wolfe algorithm, it is re-initialized: for all nodes with little flow ($x_i < 0.9$), a value of 0.4 is given to the multiplier λ_i.

Table 1: Comparison of computation times for the column generation and augmented Lagrangian methods.

		AM		PM	
n	Δ	COL. GEN.	AUG. LAGR.	COL. GEN.	AUG. LAGR.
48	0	9[1]	6[2](0)[3]	16	6 (0)
	5	19	11 (0)	33	13 (0)
	10	38	17 (11)	52	17 (0)
	15	58	36 (19)	64	59 (32)
	20	100	38 (7)	85	28 (4)
100	0	26	34 (0)	55	84 (59)
	5	38	20 (12)	73	72 (22)
	10	48	82 (41)	118	79 (69)
	15	89	50 (0)	157	67 (17)
	20	116	89 (0)	252	90 (0)
151	0	94	109 (48)	283	243 (99)
	5	170	207 (97)	358	285 (140)
	10	382	240 (117)	673	268 (206)
	15	251	227 (137)	894	388 (178)

1 time in second: to optimally solve the linear set partitioning problem by column generation.

2 time in seconds required to activate the stopping rule by the augmented Lagrangian method.

3 time required to find the best lower bound by the augmented Lagrangian method.

Table 1 shows the results with computation times in seconds on a CDC173. The time required to optimally solve the linear set partitioning problem by column generation is compared with the times required to reach the stopping rule and the best lower bound using the augmented Lagrangian method. (A zero value indicates that the network bound is the best). The overall performance of the Augmented Lagrangian method is good. Compared with the time required for the primal column generation algorithm, on average only 30% of this time is necessary to reach the best lower bound with the dual algorithm. However, another 40% of this time has to be added for the dual algorithm to reach the stopping rule. Further research needs to be carried out on this aspect of the problem.

In addition to the network structure (for the SPTW subproblem) the column generation algorithm requires considerable additional space for the simplex method, while the augmented Lagrangian method only needs space for the network structure. With the calculation of the lower bound LB at the first node of a branch-and-bound algorithm based on the network relaxation of the original problem, these LB units of flow can be forced onto the network thus reducing the number of infeasible routes and the size of the tree to be explored. The method can therefore be expected to solve larger problems to optimality.

7. BIBLIOGRAPHY

(1) Bertsekas, D.P., Constrained Optimization and Lagrange Multipliers Methods (Academic Press, New York, 1982).

(2) Desrosiers, J., Pelletier, P., Soumis, F., Plus court chemin avec contraintes d'horaires, RAIRO, Recherche opérationnelle 17(4), (1983) 1-21.

(3) Desrosiers, J., Soumis, F., Desrochers, M., Routing with Time Windows by Column Generation, Ecole des Hautes Etudes Commerciales de Montréal, #G-83-16, to appear in Networks.

(4) Desrosiers, J., Soumis, F., Desrochers, M., Sauvé, M., Routing and Scheduling with Time Windows Solved by Network Relaxation and Branch-and-Bound on Time Variables, in: J.-M. Rousseau (ed), Computer Scheduling of Public Transport, vol. II (North-Holland, Amsterdam, 1984).

(5) Desrosiers, J., Soumis, F., Sauvé, M., Lagrangian Relaxation Methods for Vehicle Routing and Scheduling with Time Windows, Ecole des Hautes Etudes Commerciales de Montréal, #G-83-26.

(6) Fisher, M.L., The Lagrangian Relaxation Method for Solving Integer Programming Problems, Management Science 27 (1981), 1-18.

(7) Geoffrion, A.M., Lagrangian Relaxation and its Uses in Linear Programming, Mathematical Programming Study 2 (1974), 82-114.

(8) Held, M., Wolfe, P., Crowder, H., Validation of Subgradient Optimization, Mathematical Programming 6 (1974), 62-88.

(9) Magnanti, T.L., Shapiro, J.F., Wagneur, M.H., Generalized Linear Programming Solves the Dual, Management Science 11 (1976), 1195-1203.

POOR

QUALITY

ORIGINAL